User Manual

PROPHET

A free NONMEM User Interface
for UNIX / LINUX

by Dirk Zeumer

Boehringer
Ingelheim

1 Table of Contents

2 Introduction

2.1 About PROPHET

PROPHET is a package of tools developed by Boehringer Ingelheim Pharma GmbH & Co. KG (BI Pharma) with the intention to replace the NONMEM standard user interface and to offer additional benefits during the process of a population pharmacokinetic / pharmacodynamic evaluation of clinical studies with NONMEM.

PROPHET offers the following functionality:

- Full NONMEM integration
- Simultaneous execution of NONMEM calculations in the same directory
- Creation of a report file providing most important information for assessment of a run in a condensed way (see 5.3 and 6.1)
- Use of 'descriptors' in input and report files for easy documentation (see 4.1 and figure 1)
- Log file creation, containing all information and data of a NONMEM calculation (see 5.4)
- Configurable graphic generator for automatic generation of (i.e.) Goodness-of-Fit plots (see 5.5 and 6.2)
- Automatic use of subdirectory structures for table, GoF, error and log files in order to avoid "overcrowded" directories (see 5.4)
- Basic file protection to avoid accidental overwriting of result files (see 5.2)

2.2 Terms of Usage

The software and accompanying written material are provided "as is" without warranty of any kind. Further, BI Pharma GmbH & Co. KG does not warrant, guarantee, or make any representations regarding the use, or the results of use, of the software or written material in terms of correctness, accuracy, reliability, currentness, or otherwise. The entire risk as to the results and performance of the software is assumed by you. If the software or written materials are defective to you, and not BI Pharma GmbH & Co. KG or his agents, or employees, assume the entire costs of all necessary serving, repair, or correction.

There are inherent dangers in the use of any software and BI Pharma GmbH & Co. KG cautions you to make sure that you completely understand the potential risks before installing and/or using the software. You are solely responsible for adequate protection and backup of the data and equipment used in connection with the software.

The above is the only warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose, that is made by BI Pharma GmbH & Co. KG, on this software. No oral or written information or advice given by BI Pharma GmbH & Co. KG, his agents or employees shall create a warranty

2.3 Further Developments / Additions to PROPHET

PROPHET is completely written in PERL. Therefore, further development and/or additions to the software are relatively easy to implement. Those further developments and/or additions are highly encouraged by BI Pharma GmbH & Co. KG as far as the following points are considered:

- please clearly state that the "new software" is based on PROPHET but is not (!) the original PROPHET itself.
- please notify BI Pharma GmbH & Co. KG about such further developments and/or additions (send an email to: Dirk.Zeumer@bc.boehringer-ingelheim.com).

3 Installation

This chapter describes the installation of PROPHET on your system. It is kept as simple as possible and reduces the user interaction to a minimum. Simply follow the instructions below.

3.1 Prerequisites

The software requirements to be fulfilled to install PROPHET are:

- Unix or Linux as operating system (OS)
- Perl 5.6 (or higher) – should be already installed on every Unix / Linux OS
- NONMEM

additionally for the use of the automatic graphic creation:

- S-Plus 5 (or higher)

3.2 Running the Installation

- Create a new directory to install PROPHET into.

  Depending on the intended number of users of PROPHET on your system this directory can be located either in your home directory (single user) or in a free accessible directory (usually `/usr/share`).

  Note: It is recommended to install PROPHET only once on each system!

- Copy the file `PROPHET.tar.gz` into the newly created directory.

- Open a command shell, move into the directory and unpack the file using the following commands:

  ```
  gunzip *.gz
  tar xvf *.tar
  ```

- Enter the following command:

  ```
  perl installer
  ```

  The installation will start immediately. In some cases the installation routine will ask for some input (usually asking for the position of some files or directories). Please enter your answer and press return for each question.

  Note: It is possible to stop the installation at every time pressing the `CTRL-C` combination on your keyboard. The installation can be restarted as often as necessary.

General note: In order to be accessible throughout the system, PROPHET needs "to be found" by the system resources. Usually this is done by adding the installation directory to the user's `PATH` variable.
The easiest way to do this is to edit the user's environment set up file that resides in the user's home directory (called `.profile`). As this file can have different syntax from user to user (depending on system and used shell type) no further explanation can be given here. Please contact your local system administrator or browse your system documentation.

3.3 Checking the Installation

Beyond the installation directory an additional directory (called `test`) was installed. It contains a NONMEM control stream (`run001.inp`) and a data file (`data1.dat`) that can be used to test the installation as following:

- Enter the directory on the shell

- Enter the following command:

  ```
  ../nmSub run001.inp
  ```

  Note: It is assumed that you don't have added the installation directory to your `PATH` variable so far. Otherwise you may skip the "`../`" in the command.

- If PROPHET was successfully installed, four new directories (`ERRORS, LOGS, TABLES, GOF`) and a new file (`run001.rep001`) have been created.
  If during the installation you have chosen to use the automatic graphic creation routine you can also find a file called `GOF001.ps` beyond the newly created `GOF` directory, indicating that the link between PROPHET and the S-Plus package is working correctly.

4 PROPHETs Conventions

In order to work in the intended way, PROPHET requires some conventions to be fulfilled regarding the NONMEM control stream. These conventions are detailed in the subchapters below.

4.1 Descriptors

In order to create a complete report that can be used as a reference for a finished NONMEM calculation, PROPHET expects additional details within the control stream – so called descriptors. A descriptor is implemented as a NONMEM comment, consisting of a keyword and a "content" or "value", separated by a ':'.

Syntax example: `;KEYWORD: Value`

All descriptors are expected to be found within the control stream before the appearance of the `$PROBLEM` statement and with a maximum length of 60 characters for their content. Most descriptors are mandatory for PROPHET but some are used as commands (see below).

With two exceptions (`NOTES` and `PLOT`) all descriptors should be used only once as the second use of a descriptor will "override" the content of the first.

The following descriptors are implemented in this version:

Keyword	Value	Description	Mandatory
`PROJECT`	Project name	Name of project to which this control stream belongs	Y
`STUDY`	Study number	Number of study to which this control stream belongs	Y
`RUN`	Run number	Number of control stream	Y
`KINETICIST`	Name or initials of pharmacokineticist	Name or initials of pharmacokineticist that defined this control stream	Y
`NOTES*`	Free text	Any comment to this control stream	N
`NOGOF`	--	Deactivates the automatic generation of the GOF plot for this control stream.	N
`NOREPORT`	--	Deactivates the automatic generation of the report for this control stream	N
`PLOT`	Plot name	Activates the automatic generation of the given plot for this control stream**	N

* : this descriptor may appear as often as needed. Content will be merged in the report.

** : the value of the descriptor has to match the defined name in the plot configuration file (see *Adding Standard Plots*)

Note: Please don't use a ':' within the value of a descriptor – otherwise the descriptor value won't completely appear in the report file.

4.2 Initial Values

In order to make the automatically created report as readable as possible, PROPHET offers the possibility that the user will give reference names (10 characters max.) to the variables used in the control stream.

As these reference names are implemented as NONMEM comments, PROPHET expects that every initial value for a variable is placed in a single line (regardless if you are using reference names or not!). The reference name is placed behind the initial value, separated by a ';' from it.

Example:

```
$THETA
      (0, 0.02, )  ; CL
      100          ; V2

$OMEGA
      BLOCK(2)  0.1   ; Eta_CL
                0.1   ; Corr_CL/V2
                0.1   ; Eta_V2
```

4.3 Table Statements

PROPHET requires that all tables files that are created by the use of the `$TABLE` statement in your control stream are ending with the letters "tab".

Only table files corresponding to this requirement will be automatically transferred from the temporary working directories into the subdirectory structures and will receive the basic result file protection.

4.4 Plot Generation

PROPHET will automatically call its graphic generation routine after the calculation of a NONMEM control stream (unless you aren't using S-Plus or have used the `NOPLOT` descriptor in your control stream).

In order to create the Goodness-of-Fit (GoF) plot from the results, PROPHET expects a certain kind of table as an output file from your calculation. This table file is called `sdtab` and is identical to the file specified by the S-Plus library Xpose.

From the Xpose 2.0 User's Manual (page 7):

"`sdtab` stands for standard table file. It should contain items describing the overall goodness of fit. The recommended column items and order for this table file is `ID, TIME, IPRED` and `IWRES` (in addition to these NONMEM by default adds `DV, PRED, RES` and `WRES`). `IPRED` (individual predictions) and `IWRES` (individual weighted residuals) are not NONMEM defined items and has to be defined by the user. These can be obtained by the following code in the `$ERROR` block:

```
$ERROR
  IPRED = F
  W     =    ; Your choice:
             ;   1                = additive error model
             ;   F                = constant CV error model
             ;   F**THETA(.)      = power error model
             ;   (F**2+THETA(.)**2)**0.5
             ;                    = additive plus
             ;                      proportional error model

  IRES  = DV-IPRED
  IWRES = IRES/W
  Y     = IPRED + W*EPS(1)
```

Note that the `IWRES = IRES/W` line will, in some cases, need reformulation to avoid division by zero."

This reformulation can be done by adding the following code (replacing the original `IWRES=IRES/W` line from above):

```
DEL = 0
IF(IPRED.EQ.0) DEL = 0.001
IWRES = IRES / (W + DEL)
```

Important note: When coding your `$INPUT` statement, please make sure that your dependent variable is called "`DV`" only (not "`CP=DV`" or something similar)!
Otherwise NONMEM will add your dependent variable with its "original" name into the table file and the automatic graphic generator will fail (as it expects a DV column).

5 Working with PROPHET

5.1 Starting a NONMEM calculation

Once a control stream has been created it can be submitted with the following command (on the command shell):

```
nmSub name_of_file
```

Before passing the file to the NONMEM package, PROPHET will check the correct use of the descriptors and report any error on this topic on the screen (in this case further processing will be stopped). Further internal checks will be done in the background. If all checks have passed, the calculation will be started.

5.2 Actual Run Number & File Protection

PROPHET will try to protect previous results from being overwritten when a user submits a control stream again without changing the file name (i.e. after changing the initial values). This will be done be adding the so called "actual run number" (ARN) as extension to the file names of all results (including table files!).

In the beginning, the ARN equals the value of the `RUN` descriptor. Every time a control stream is submitted, PROPHET checks the existence of a log file for this control stream that ends with this value. If such a file exists (meaning that the control stream has been submitted before) PROPHET appends a ".x" to the ARN (and therefore to the file name). This 'x' is increased until a new, unique filename can be build.

Example:
Given a control stream name of "`run001.inp`" and a `RUN` descriptor value of "`001`", the ARN will be:

# of submissions	ARN
1st	001
2nd	001.1
3rd	001.2
…	

5.3 Result Directories

In every directory from which `nmSub` is called, PROPHET automatically creates a set of subdirectories in which the result files of every calculation will be stored. These directories and their use for PROPHET are as follows:

`ERRORS`	contains all error files
`GOF`	contains all GoF plots

`LOGS`	contains all log files
`TABLES`	contains all result table files

The report file is stored in the same directory as the control stream. An example of a report and a standard GoF plot can be found in 6.1 and 6.2.

5.4 Result Files

A control stream calculation will result in up to 4 result files and the table files specified in the $TABLE statements:

Kind of File	File Extension*	Content
log file	.log[ARN]	ASCII file containing all information about the calculation and its results in a single ASCII file (i.e. control stream, reduced data file, NM-TRAN output, NONMEM compile script, NONMEM output).
report file	.rep[ARN]	ASCII file containing a condensed report of the NONMEM output.
error file	.err[ARN]	ASCII file containing all relevant warning and error messages from the calculation (normally NM-TRAN output only). The “default” warning (“`NM-TRAN infers that the data are population`”) is not recorded.
GoF file	[ARN].ps	PostScript file with the standard GoF plot. Needs a postscript viewer to be displayed on screen (like Ghostview).
table file(s)	tab[ARN]	ASCII file(s) containing the information asked for in the $TABLE statement of the control stream.

* : the expression [ARN] will be replaced by the value of the actual run number

5.5 Reduced Data File

In order to use as less disk space as possible PROPHET will reduce the given data file for the calculation to the necessary minimum. This new data file, called “reduced data file” contains only those columns that are needed by NONMEM, skipping all columns that are dropped or not referred to in the `$INPUT` statement of the control stream (the `$INPUT` will be automatically adjusted).

The reduced data file and the changed control stream will be used for calculation only (the original files remain unchanged!) and will be stored in the log file.

5.6 Adding Standard Plots

The automatic plot generator offers the possibility to include user written S-Plus scripts to the set of available plots. These plots need to be specified in the plot configuration file (`nmPlot.cfg`) residing in the installation directory.

An entry in the plot configuration file has the following structure:

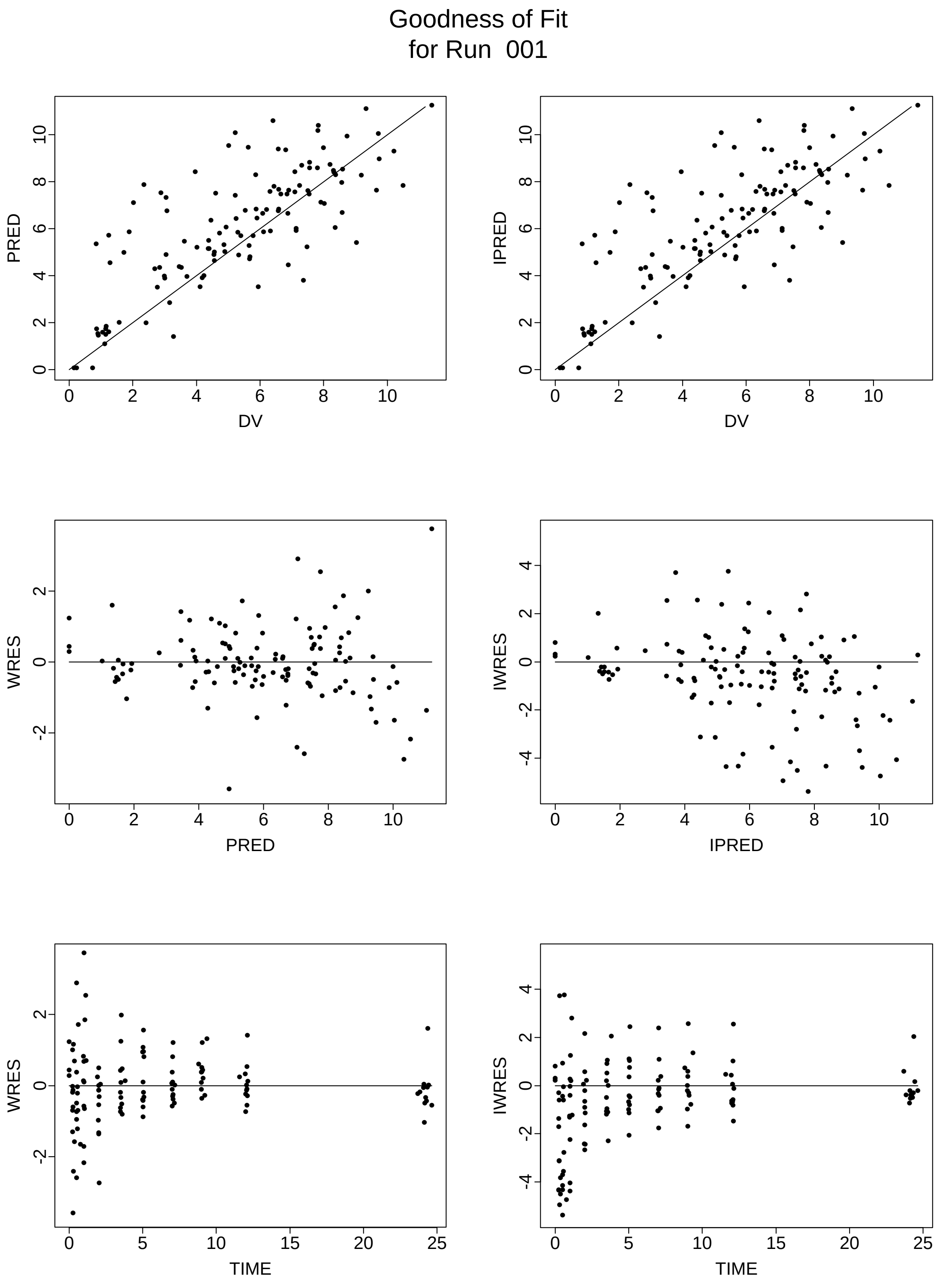
`[PLOT_NAME]`	Name that will be used as value for the PLOT descriptor in the control stream (case sensitive!).
`PLOT = /full/path`	Complete path to the script that should be run by S-Plus.
`DEPEND = table_name`	Name of table file(s) required by the script. Multiple file names have to be separated by a single blank. If any of the files are missing, the plot routine will not be called.
`TARGET = directory`	Name of subdirectory to which the plot will be stored.

6 Examples

6.1 Example of Report

```
--------------------------------------------------------------------------------

                         Report for file : run001.log001

--------------------------------------------------------------------------------
Project : Theophylline         Study : 0815 - PROPHET Test             Run : 001

Notes : Adapted version of the CONTROL5 file from NONMEM
--------------------------------------------------------------------------------
Kineticist : Demo User
Input File : run001.inp
Data File  : THEOPP_red

--------------------------------------------------------------------------------
Subroutines : ADVAN2                                  Method     : FO
Records     : 144         Observations : 132          Subjects   : 12
Obj-Func.   : 104.561     Evaluations  : 193          Sig.Digits : 4.2
--------------------------------------------------------------------------------
Gradient:
   .8295E-02    .2714E-01   -.5259E-01    .2081E-02   -.3910E-03    .9792E-03
   .4103E-03    .1779E-02    .1210E-02   -.4317E-02

0MINIMIZATION SUCCESSFUL

--------------------------------------------------------------------------------
Parameter
     Description  Initial       Estimate    STDerr     %SE     95%-Interval
                                                              (EST +/- 2x STDerr)
THETA
 1 : KA           (.1,3,5)        2.7700    0.7080   25.56    1.3540    4.1860
 2 : K            (.008,.08,.5    0.0781    0.0073    9.30    0.0636    0.0926
 3 : CL           (.004,.04,.9    0.0363    0.0045   12.48    0.0272    0.0454

OMEGA
 1 : ETA_KA       6               5.5500    4.8300   87.03   -4.1100   15.2100
   : CORR_KA/K    .005            0.0052    0.0140  267.18   -0.0228    0.0332
 2 : ETA_K        .0002           0.0002    0.0001   50.83   -0.0000    0.0005
   : CORR_KA/CL   .3             -0.1280    0.4740 -370.31   -1.0760    0.8200
   : CORR_K/CL    .006            0.0091    0.0037   40.83    0.0017    0.0166
 3 : ETA_CL       .4              0.5150    0.2120   41.17    0.0910    0.9390

SIGMA
 1 : add. Error   .4              0.3880    0.1050   27.06    0.1780    0.5980
--------------------------------------------------------------------------------

$INPUT ID DOSE=AMT TIME DV WT

$PK
   CALLFL=1
   KA=THETA(1)+ETA(1)
   K=THETA(2)+ETA(2)
   CL=THETA(3)*WT+ETA(3)
   SC=CL/K/WT

$ERROR
   IPRED=F
   DEL=0
   IF(IPRED.EQ.0) DEL=0.0001
   W=1  ; additive error
   IRES=DV-IPRED
   IWRES=IRES/(W+DEL)
   Y=IPRED+W*EPS(1)

$COV

$TABLE ID TIME IPRED IWRES ONEHEADER NOPRINT FILE=sdtab001

--------------------------------------------------------------------------------
```

6.2 Example of GoF Plot


Goodness of Fit
for Run 001
PRED
DV
IPRED
DV
WRES
PRED
IWRES
IPRED
WRES
TIME
IWRES
TIME